GeSSi: vers um simulateur d´environnements Peer-to-Peer Générique et Scalable ?

Nicolas Maillard
CPAD/HP - PUCRS
ID - 18 décembre 2003

Plan

- Introduction:
  - Brésil, Porto Alegre, PUC-RS, ...
  - CPAD
- Calcul P2P et Simulation
- GeSSi
  - Architecture
  - Spéc. de scénarios
- Résultats expérimentaux
- Conclusion

Couleur locale...

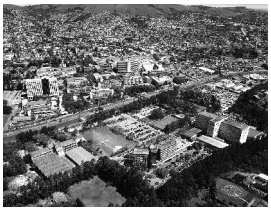
Couleur locale...

Couleur locale...

Couleur locale...

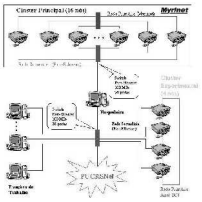
PUCRS - Porto Alegre

- 55 ans.
- Faculté d´Informatique 1000++ élèves, 68 profs.
- Master (bac+5 :o) - 55 élèves



CPAD (1)

- Centre de Calcul Haute Performance
- 3 clusters
- 1 Prof. (César De Rose), 1 post-doc, 2 Masters, 6 étudiants
- 1 sponsor: HP



CPAD (2)

Thématiques:

- Architecture Itanium (Gelato)
- Gestion de jobs sur cluster (PBS/OAR-like): Crono
- "Grappe Virtuelle": vCluster
- Algorithmes de Gestion de Ressources Distribuées (DPM)

Plan

- Introduction:
  - Brésil, Porto Alegre, PUC-RS, ...
  - CPAD
- Calcul P2P et Simulation
- GeSSi
  - Architecture
  - Spéc. de scénarios
- Résultats expérimentaux
- Conclusion

Grid et P2P Computing

- Distinction entre les deux ?
  - Égalité entre les noeuds (=! client/serveur) ?
  - Sécurité vs. anonymat ?
- Structuration et auto-apprentissage
  - émergence d´une structure de Grille au sein d´une structure P2P

P2P : beaucoup de choses...

- **Gnutella** : chaque noeud = *servent*. Bcast des requêtes. Time-to-Live.
- **Freenet** : Mémoire LRU de voisins + fonctions de hachage pour guider les requêtes. Time-to-live.
- **I-cluster** [Richard] : Mémoire LRU de voisins + bavardage (non-déterminisme)
- **Chord** [Stoica] : spéc. d´une fonction de hachage (hachage consistant) distribuée pour distribuer les clés.
- **Super-Peers** [Garcia-Molina, Stanford] : hiérarchie entre les peers.

... et peu de validation !

- Analyse de Réseaux P2P existants:
  - Gnutella [Yang&Garcia-Molina 2002, Oram 01]
- Simulation
  - Freenet [Oram 01, "Performance"]
  - Chord [Stoica 02]
- + des simulateurs pour les Grilles !
  - GridSim [Buyya]

Simulateurs

- **Aurora**: C, séquentiel, 400 000 noeuds simulés [Performance]
- **Chord**: C++, séquentiel, API pour la spécification des scénarios, 100 000 noeuds simulés.

- **Simgrid** [Casanova]: C, ordonnancement de tâches. API pour description des tâches et des ressources.
- **GridSim**: Java (SimJava) + threads, haut-niveau: ressources = grappes, MPP,... 5000 "PEs"
- **Platus** [Bernstein, Dotti]: Java + threads. Vérification formelle de la correction des algos. distribués.

Tous reposent sur la gestion d´événements notifiés par un ordonnanceur central.

Simulateurs

Haut-Niveau, 1000 noeuds

- **Aurora**: C, séquentiel, 400 000 noeuds simulés [Performance]
- **Chord**: C++, séquentiel, API pour la spécification des scénarios, 100 000 noeuds simulés.

Hacké et efficace!

- **Simgrid** [Casanova]: C, ordonnancement de tâches. API pour description des tâches et des ressources.
- **GridSim**: Java (SimJava) + threads, haut-niveau: ressources = grappes, MPP,... 5000 "PEs"
- **Platus** [Bernstein, Dotti]: Java + threads. Vérification formelle de la correction des algos. distribués.

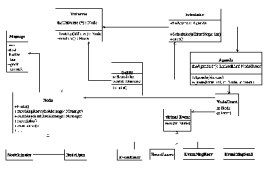
Tous reposent sur la gestion d´événements notifiés par un ordonnanceur central.

Plan

- Introduction:
  - Brésil, Porto Alegre, PUC-RS, ...
  - CPAD
- Calcul P2P et Simulation
- GeSSi
  - Architecture
  - Spéc. de scénarios
- Résultats expérimentaux
- Conclusion

GeSSi: Generic Scalable Simulator

- Écrit en Java
- Partie générique = 19 classes
  - Universe = ensemble de Nodes
  - Events
  - Scheduler = Agenda de couples (node, event)
  - Stats



GeSSi (2)

+ parties spécificiques pour chaque algo simulé

- Spécialisation, héritage, classes dédiées (mém LRU, voisins, événements,...)
- 3 algos spécifiques: DPM, Icluster, FreeNet (en cours)

L´utilisateur doit:

- définir par héritage ses événements
- programmer NodesXXX extends Nodes pour définir ce que les noeuds font.

Node

- 1 identificateur (adresse IP)
- Des voisins (références sur d'autres noeuds)
- Des méthodes
  - send(Message m)
  - recv(Message m) (+ bcast())
  - (*Ça devrait être des méthodes d'une classe Network !)*
- Un ensemble de méthodes execXXX() = ce qu'un noeud doit faire quand un événement survient.

Messages

- Un noeud source, un noeud dest
- Un tag
- Un ttl
- Pack() et unpack() pour différents type (int, Node, list<Node>)

Event

- Classe abstraite d'ou dérivent les événements spécifiés par l'algo

- Méthode exec(Node n) à implémenter:

```
public class EventEnter extends Event
{
  public String name = "enter" ;
  public void exec(Node n)
{   n.execEnter() ; }
}
```

Universe

- = ensemble de tous les Node du système.

- Implémenté comme une table de hachage
  - Node lookup(int idf)
  - Node random()

- Itérateurs

- Méthodes statistiques (meanDegree, [illegible] minimum)

Scheduler

- = un Agenda de n timesteps
- Agenda = table de liste de couples (Node, Event)
- Une méthode exec() qui parcourt la table:

```
for (int i= 0 ; i< numHopsMax ; i+ + ) {
      NodeEvent ne =  Agenda.dequeue(i) ;
        while (ne != null) {
        Event e = ne.theEvent ;
        Node n  = ne.theNode ;
        e.exec( n ) ;
        ne =  Agenda.dequeue(i) ;
      }
```

Scénarios

- = fichier ASCII utilisé pour le constructeur du Scheduler
- API simple pour l'utilisateur

```
STRATEGY Icluster     ## Nom de l´algo
TOTAL_NODES 20000   ## Nombre de noeuds
TIME_STEP_SIZE 1      ## durée (min) du timestep
SIMULATION_TIME 43500  ## durée simu (1 mois)
10 23 1 1 * Enter [1-4],[20-50],[300-1000] ### events
00 12 * * * Leave r[0-19999]
```

- Fuseaux horaires
- Lois de distribution
- Création automatique de scénarios (fuseaux)

Plan

- Introduction:
  - Brésil, Porto Alegre, PUC-RS, ...
  - CPAD
- Calcul P2P et Simulation
- GeSSi
  - Architecture
  - Spéc. de scénarios
- Résultats expérimentaux
- Conclusion

Exemple d´algorithme: Icluster

- L'algo de I-CLuster [B. Richard 2003] est basé sur le bavardage
- Chaque noeud maintient une table de voisins estampillés
- A intervalles fixes, tirage aléatoire d´un voisin et échange/fusion des tables de voisins

- Les infos diffusent en temps logarithmique.

- Mécanisme de liste de suspect pour détection des fautes.

- Scénario basé sur l´analyse de l´intranet HP

Quel scénario ?

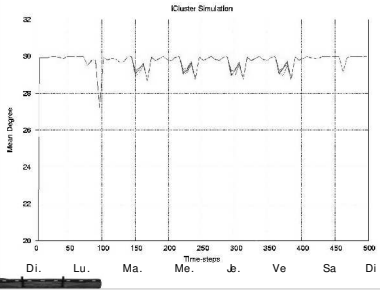
2 scénarios “jouets”:
- 3 “fuseaux” (US, Amérique du sud, Europe)
  3 types de noeuds
  - serveur (tout le temps on-line)
  - Station de travail (08h00 - 12h00, 14h00 - 18h00)
  - PC domestique (18h00, 23h00 + we)

  Distribution des noeuds ?

- 4 fuseaux, 1 type de noeuds avec un profil donné [G. Pezzi].

1 semaine, 20 000 noeuds

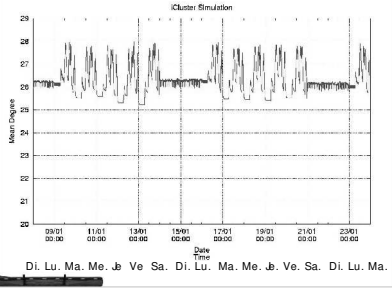
1 timestep = 20 min - simu = 10 min (P4 1.7 GHz 400 Mo)



2 semaines, 20 000 noeuds

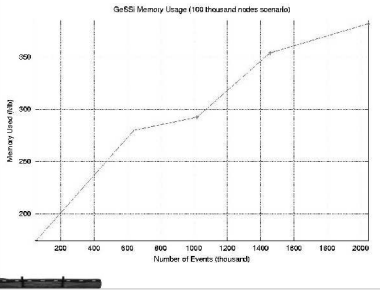
[G. Pezzi]

1 timestep = 1 min - simu = 1h30 (PIII 550 MHz 256 Mo)



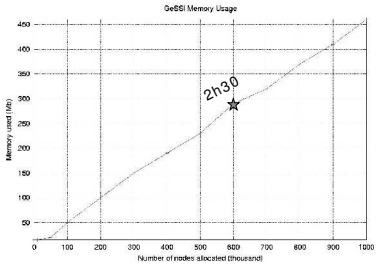
Mémoire vs. Événements

(temps simulé = 1 jour)



Mémoire vs. nb noeuds

entrée de 1000 noeuds, 1 jour



Plan

- Introduction:
  - Brésil, Porto Alegre, PUC-RS, ...
  - CPAD
- Calcul P2P et Simulation
- GeSSi
  - Architecture
  - Spéc. de scénarios
- Résultats expérimentaux
- Conclusion

Bilan

- GeSSi, ça marche...
- Générique, et “scalable” (hum ?)
- Simulation logique uniquement
- Possibilité de comparaison simple entre les algos.

À faire

- Libération dynamique de la mémoire (vers +1 Gnoeuds)
- Scénarios !!!
- Freenet !!!

- Élaboration de nouveaux algorithmes?